



# Implantando uma arquitetura de serviços RESTful no CPD/UFSM

Universidade Federal de Santa Maria

Centro de Processamento de Dados

Everton de Vargas Agilar (UnB / UFSM) Jader Adiel (UFSM)





#### Plano

- ► Barramento de serviços ERLANGMS
  - Objetivo do projeto
  - Panorama geral sobre modernização
  - Histórico do projeto
  - Design da arquitetura
- ► Implantação do barramento na UFSM
  - ► O que está sendo feito
  - Principais desafios arquitetura RESTful
  - ► Caso prático 1: Web services implementados para o NCC
  - ► Caso prático 2: TClis do SIE invocando web services Java





## Barramento de serviços ERLANGMS





## Objetivo do projeto

#### **ERLANGMS**

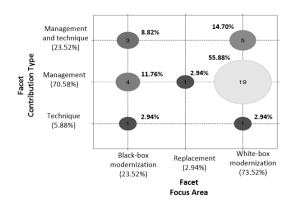
É uma abordagem desenvolvida na UnB para facilitar a modernização de sistemas para uma arquitetura orientada a serviços por meio de um barramento de serviços + SDK + processo de modernização e arquitetura documentado (SMSOC).





### Panorama geral sobre modernização

Estudo realizado para identificar as estratégias de modernização







### Histórico do projeto

2015 2016 2017 2018 2019

Primeira versão básica de um barramento REST em Erlang.

Foi escrito um SDK simples para desenvoyler servicos em Java usando os operadores filter. fields, sort e limit.

Estudo de caso para demonstrar o uso da REST API

O barramento recebeu suporte para HTTP 2.0 e HTTPS

aohsh

A interface da API foi melhorada com a possibilidade de realizar filtros mais complexos.

A capacidade de serialização foi ampliada para tratar melhor os tipos de

A API passou a suportar operadores de atributo.

Foco em desempenho com a Implementação de cache de queries usando named queries da JPA.

Implementação do ResultCache para cachear o resultado do webservice no harramento e evitar que requisições iquais seiam enviadas para o servidor de aplicação.

O SDK recebeu diversas melhorias de performance e estabilidade

Foco na segurança com suporte para dois protocolos: OAuth2 e I DAP v3

Foi incluído a possibilidade de gerar consultas com SQL nativo usando os operadores.

A SDK está sendo redesenhada para ser mais agnóstica e consequir trabalhar com Sprint e Java FF

O operador filter está sendo ampliado para suportar condições com and/or e precedência de condições.

Está sendo implementado suporte para API constraints para limitar o uso da API em cada endpoint.

A serialização de dados está sendo refinada para fazer a conversão de entidades e value objects (VO).





## Histórico do projeto

Principais features em uso na UnB

#### **Features**

- ► HTTP/2, HTTP/1.1, HTTPS (Secure TLS Listener);
- ► LDAP v3 Proxy LDAP;
- ► HTTP Basic authentication;
- OAuth2 authentication;
- SDK para criar serviços em Java;
- ► Query Api filter, fields, sort, limit.
- ► Suporte para Dados Abertos.





## Design da arquitetura

#### Design da arquitetura

- Multiplataforma e arquitetura modular;
- Serviços especificados em catálogos de serviços;
- Cluster de serviços para evitar ponto único de falhas;
- Suporta servi
  ços RESTful por contrato;
- ► Modelo de concorrência para serviços em Erlang: Actor Model;
- ► Baixa curva de aprendizado.





## Implantação do barramento na UFSM





## O que está sendo feito

#### O que já foi feito

- Implantação do barramento em partes, priorizando:
  - Query API
  - Subsistema de serialização;
  - Adaptar a interfaces do SDK para trabalhar de forma agnóstica com Java EE e Spring.
- ► Features visando a migração de backends SIE para Java:
  - ► Novo operador format para serializar dados como Dataset;
  - Novo subsistema provider no SDK para permitir seu uso independente da camada de negócio;
  - Provider para trabalhar com meta-queries do Delphi;
  - Cliente REST implementado no Delphi.





### O que ainda precisa ser feito x prioridade

#### O que ainda precisa ser feito

- Tornar a camada de fachada mais simples Moderado;
- Implementar o subsistema de autenticação OAuth2 (ou integrar por completo o barramento que já possui) – Alto;
- Analisar os principais métodos do TCliBusiness e prover um suporte adequado no SDK para facilitar a migração – Alto;
- Permitir gerar uma consulta por meio da Query API para buscar dados de um atributo lista de um objeto principal – Baixo;
- ▶ Permitir encadear funções nos atributos das entidades com o operador % para modificar a forma como o dado é lido da fonte de dados – Baixo.





## Principais desafios arquitetura RESTful

#### Principais desafios arquitetura RESTful

- Basicamente, a maior parte do backend Delphi é um desafio levando em consideração o custo de tempo para extrair as regras de negócios comuns automatizadas no core do SIE. Dessa forma, estamos preferindo focar mais nas TClis dos frontends;
- ► Há exceções, as meta-queries do Delphi podem ser convertidas facilmente em webservices como o SDK do barramento;
- ► A quantidade de herança e, principalmente, métodos virtuais nas classes bases do pkCDESP tornam a migração mais demorada;





## Principais desafios arquitetura RESTful

#### Principais desafios arquitetura RESTful

- Para o cliente REST do frontend Delphi obter os dados no formato Dataset para os TClis, é necessário especificar o schema dos dados compatível com o esperado o que é fácil mas trabalhoso;
- ▶ Pelo menos por enquanto, uma das maiores vantagens em se utilizar microserviços é meio que perdida, visto que o Sie-Core é um artefato monolítico em termos de unidade de deploy.





# Caso prático 1: Web services implementados para o NCC







# Caso prático 2: TClis do SIE invocando web services Java







#### Trabalhos relacionados

- ► A systematic mapping study on legacy system modernization 2016
- Uma abordagem orientada a serviços para a modernização de sistemas legados – 2016
- Uma Implementação do Protocolo OAuth 2 em Erlang para uma Arquitetura Orientada a Serviço – 2017
- ► An experience report on the adoption of microservices in three brazilian government institutions 2018
- ► Melhoria na Publicação de Dados Abertos: Automatização na Publicação e Indexação Semântica dos Dados — 2018
- ► Integração das Bases de Login e Senha dos Sistemas da Universidade de Brasília - UnB - 2019
  16/17





## Obrigado!